INTRODUCTION
My research career began as an undergraduate in Software Engineering, exploring formal specification of functional models. My interest in the human aspects of software engineering led me to Human-Computer Interaction (HCI) and Ubiquitous Computing as a graduate student, where I have investigated automated meeting capture and access. I implemented and evaluated prototypes that capture several software engineering discussions and evaluated the access of those meeting records. In the future, I will continue my research in Human-Computer Interaction, Software Engineering, and Ubiquitous Computing, focusing on problems in Computer Supported Cooperative Work (CSCW) that are faced by software engineers. I plan to study the processes and tools of software engineers, as well as apply new technologies, such as ubiquitous computing, to aid in those processes. I am particularly interested in the problem of capturing and utilizing the artifacts and communications generated by work groups. I will continue to design and develop technology for initial controlled evaluations that evolves to deployable systems to study in the field in real use.

MOTIVATION
Software development teams generate a large amount of information. Teams gather requirements and create design specifications. Managers generate schedules and milestones. Quality control groups create and execute test plans. Much of this information is documented in structured, standard ways. However, as people brainstorm possibilities, sketch alternatives, evolve documents, and make decisions based on various assumptions along the way, many of these details are naturally lost in creating coherent, finished requirements, designs, and documentation. Yet some of this lost information may later be valuable to provide additional context, details, and decisions to aid in further development efforts. Documenting all of this rich, often informal information is cumbersome, and those additional costs are rarely worth the potential benefits. Finally, despite the efforts of the Software Engineering research community, there are still many projects, particularly small or informal ones, where documentation of any sort is rare and unorganized.

Thus, there remains a challenge in how to gather, link, visualize, and utilize the myriad of formal and informal artifacts, and synchronous and asynchronous communications that are generated during a project. My thesis work investigated one aspect of this problem — that of capturing and utilizing synchronous, content-rich communications, namely meetings.

MEETING CAPTURE AND ACCESS
One of the themes of ubiquitous computing is the capture and integration of everyday activities in order to provide a multimedia record of those activities for later access and use. By applying automated capture and access technologies to work discussions, large amounts of informal project information may be recorded and preserved. Capture and access involves more than simple recording. Research in this area includes investigations of the technology involved, interfaces for browsing and replaying meetings, and user requirements and social implications. It emphasizes indexing the recorded information so that review is efficient and easy, thus potentially enabling anyone to find the information they need when they need it. I have explored using the natural activities and artifacts of various meetings as indices into the multimedia record. As much of this indexing is done automatically, or as a consequence of existing processes or activities, the benefits can be realized with little or no additional time and effort of the meeting participants.

Meetings vary greatly, differing in purpose, formality, and content across domains, organizations and teams. People have difficulty envisioning how they would take advantage of captured information, and what information would be most useful to their work. As such, in my thesis work, I implemented and evaluated capture and access prototypes in three domains, each capturing a different type of meeting involving different activities and artifacts. In exploring the behavior, advantages, and disadvantages of using this cutting-edge technology, I have used both controlled experiments to study specific behaviors and extended studies of a deployed system to reveal realistic use and benefits.

TeamSpace In collaboration with Boeing and IBM Research, I explored a general notion of a meeting with TeamSpace [C6, C7]. TeamSpace is a system for managing and capturing meetings and other artifacts, such as agendas, presentations, notes and action items. Project team members can create, view, and edit these items before, during and after a meeting, and use them to navigate the recorded meetings. Thus, TeamSpace successfully integrates important project artifacts, such as action items, with synchronous communication. Each of the meeting artifacts is useful in navigating and finding information in the recording. The system has been most extensively used within a working research group for their weekly meetings for the past two years. Users have reviewed missed
meetings, practice talks, and assigned action items. While meeting review is not a very frequent activity for any individual, the ability to do so when needed is highly valued and desired.

**SAAMPad and SAAMPlayer** While TeamSpace could be used in a variety of meetings, it is not appropriate for more specific types of discussions. I investigated the capture of Software Architectural Analysis Method (SAAM) sessions, which are structured analysis meetings. I was involved in the design of SAAMPad [C3], a system that allowed users to create and manipulate software architectures on an electronic whiteboard, using those interactions as indices into the audio recording. I also implemented SAAMPlayer, a prototype access interface, using video recordings of several real analysis meetings and a summary document of those meetings. These prototypes examined the use of a specific artifact, namely the software architecture, as a captured artifact and as an index into the recorded meeting. Evaluations of SAAMPlayer showed that users were able to efficiently find information in the hours of video using multiple architectural elements as indices to find areas of interest [T4].

**TAGGER and TagViewer** TAGGER, developed in collaboration with Smart Information Flow Technologies, is a prototype system to capture requirements-gathering interviews for large development projects, called knowledge acquisition sessions [J1]. Like SAAM, this type of meeting is also fairly structured, with specific expected outcomes such as domain models and requirements specifications. However, in this case, artifacts to help structure the recording do not yet exist. Thus, the indices we gather are more user-driven — users assert software-related concepts called tags, such as domain terminology, objects, and issues, to index portions of discussion. I created a prototype access interface, TagViewer, and used it to investigate the potential benefits and uses of this method of capture in creating requirements documents [R1]. This experiment showed that the indices were used in a variety of ways to gain access to the details of a knowledge acquisition session contained within the transcript and video. In contrast, users given a non-indexed video did not use the video and relied solely on their notes of the session. Thus, this semantically meaningful indexing was crucial to enable people to successfully consult the video recording when needed, and find details they may otherwise have forgotten.

**FUTURE DIRECTIONS**

In my thesis work, I have shown the value of using a variety of indices drawn from the artifacts and activities of meetings. I am interested in continuing this work in several ways. I would like to evaluate the use and benefits of automated capture and access in additional settings over the long term. There are many interesting issues, such as how long captured material remains useful, adoption and use patterns, and privacy concerns that can be understood with longer deployments. I would also like to explore augmenting existing artifacts, tools, or development environments with captured discussions. Finally, I would like to explore the larger challenge of recording, navigating, and using the large amount of information a project can generate. Throughout my research, I have collaborated closely with a variety of researchers at Georgia Tech, Boeing, IBM Research and Smart Information Flow Technologies. The different expertise that collaborators brought to projects has been crucial to the progress of the research. I will continue to build fruitful research relationships with these and other institutions. Additionally, I am interested in collaborating with researchers outside of my areas of expertise, such as those who investigate information visualization, knowledge management, and information retrieval. My research thus far has given me a good understanding of the implementation and evaluation of interactive, collaborative, and ubiquitous computing systems, as well as an understanding of their application to software engineering problems. I look forward to advising students who will bring additional ideas and insight and take my research vision in their own directions.